

**Universidade de São Paulo**  
**Instituto de Ciências Matemáticas e Computação de São Carlos**  
**SCC-205 - Teoria da Computação e Linguagens Formais**

---

**Trabalho 1**

---

Docente:

Prof Dr. Thiago Alexandre Salgueiro Pardo

Alunos:

Luiz Eduardo Dorici	4165850
Werik Fachim	7656512
Wesley Tiozzo	8077925

Outubro de 2017

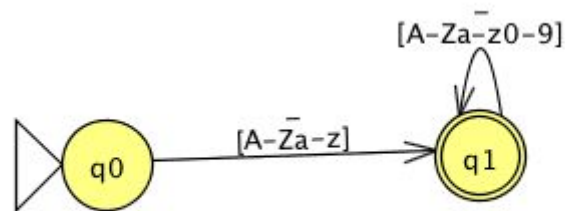


# Sumário

<b>1. Autômatos</b>	2
1.1 Identificadores	2
1.2 Números inteiros com e sem sinal	2
1.3 Números reais com e sem sinal	2
1.4 Símbolos especiais	3
1.5 Diagrama de união dos autômatos	4
1.6 Autômato determinístico finito (DFA)	5
1.7 Teste de aceitação e rejeição de cadeias	6
1.8 Minimização do autômato DFA	7
1.9 Gramática e regras de produção do autômato	8
<b>2. Expressões Regulares (GREP)</b>	9
<b>3. Gramáticas Regulares e Prolog</b>	10
<b>4. Referências</b>	12

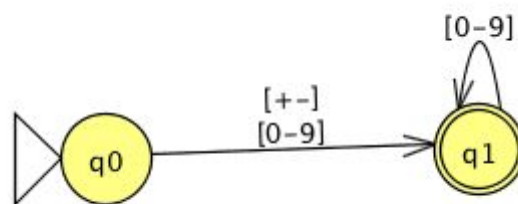
# 1 Autômatos

## 1.1 Identificadores



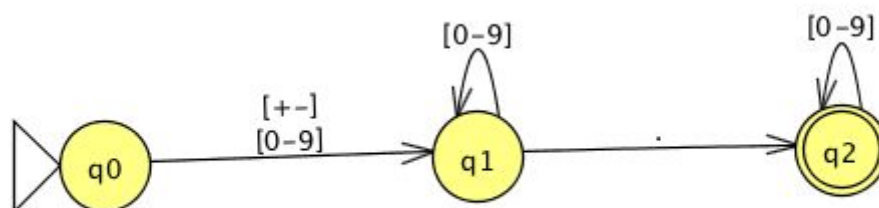
M1

## 1.2 Números inteiros com e sem sinal



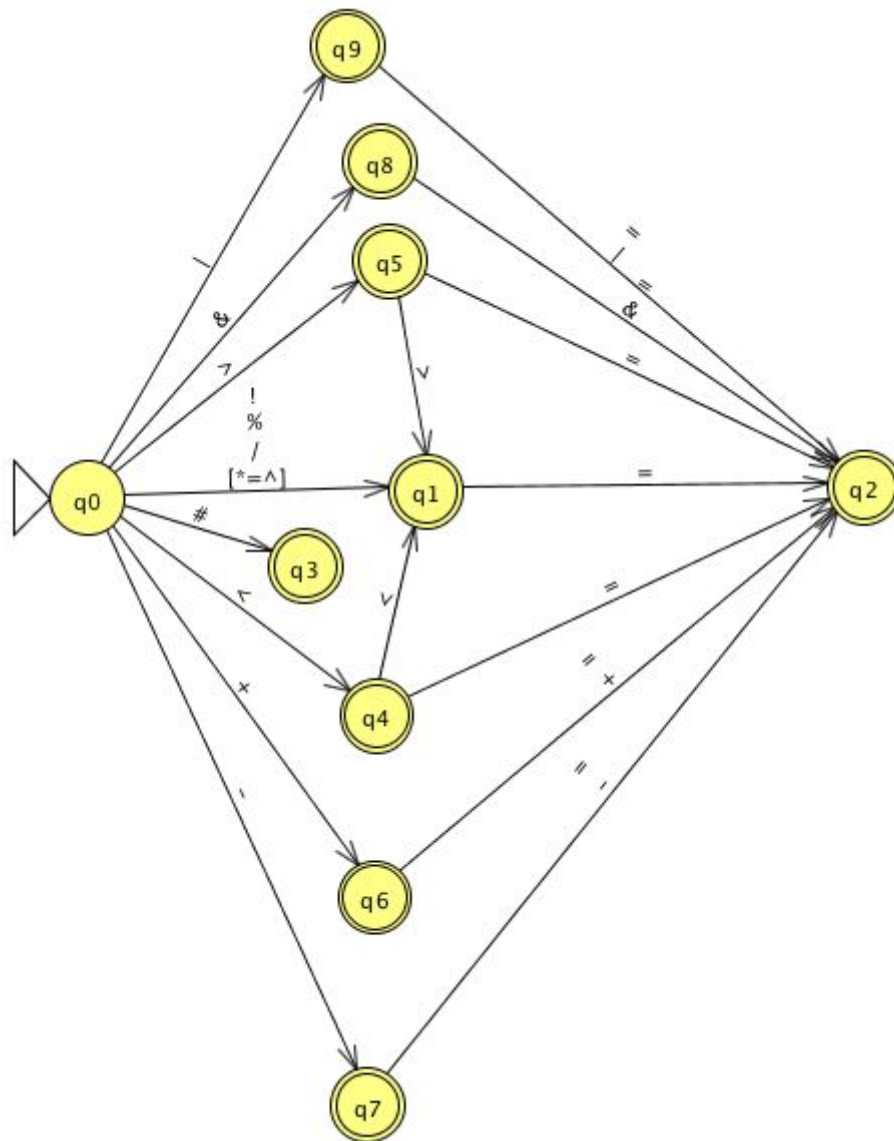
M2

## 1.3 Números reais com e sem sinal



M3

## 1.4 Símbolos especiais



**M4**

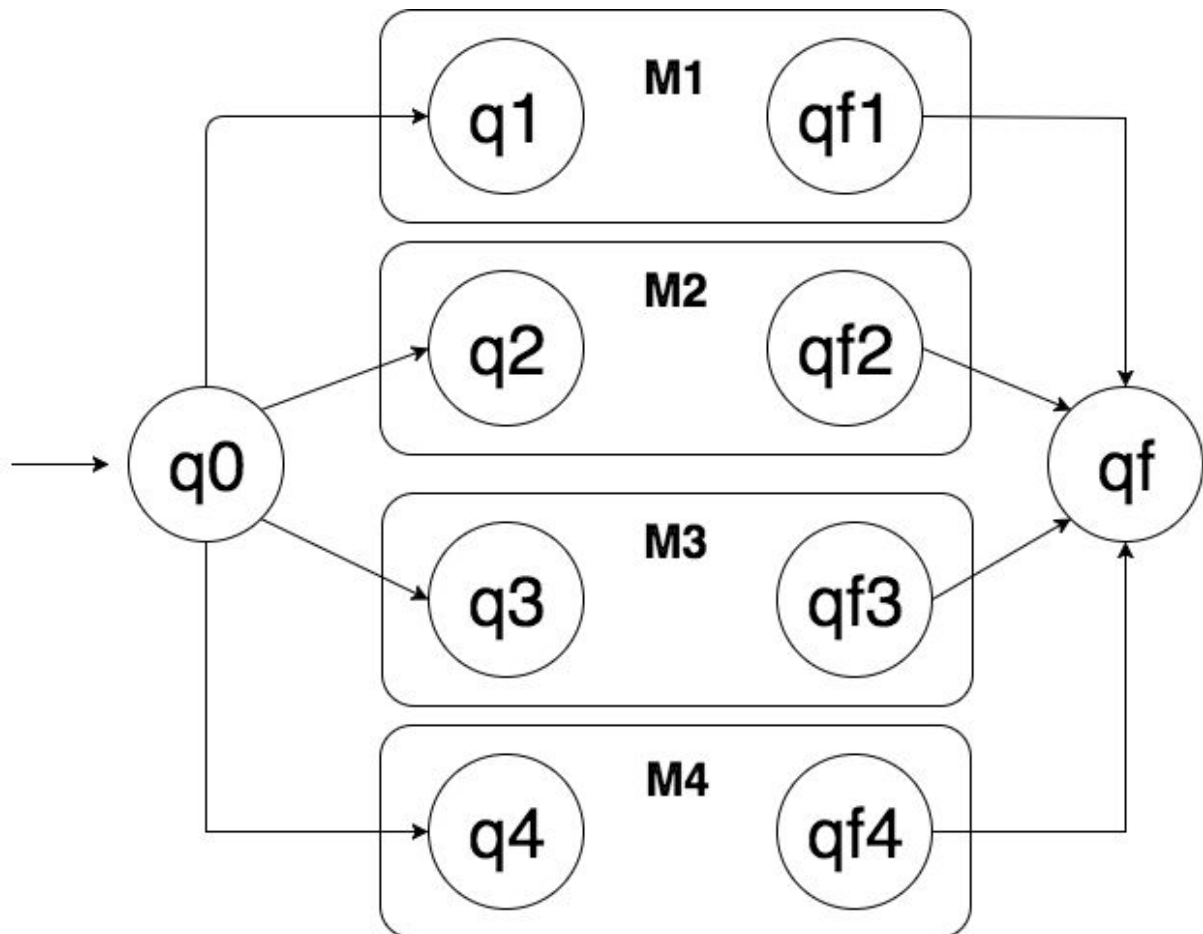
**1.5 Diagrama que representa a união dos autômatos M1, M2, M3, M4, onde:**

M1: Identificadores.

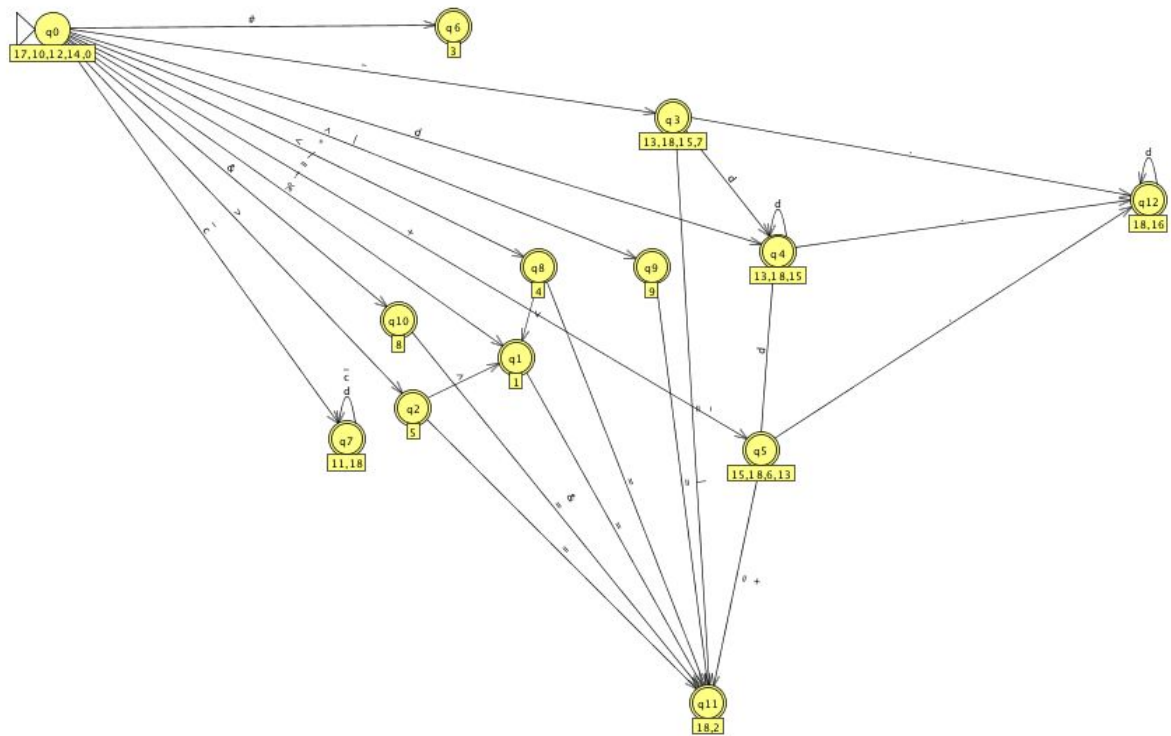
M2: Números inteiros com e sem sinal.

M3: Números reais com e sem sinal.

M4: Símbolos especiais.



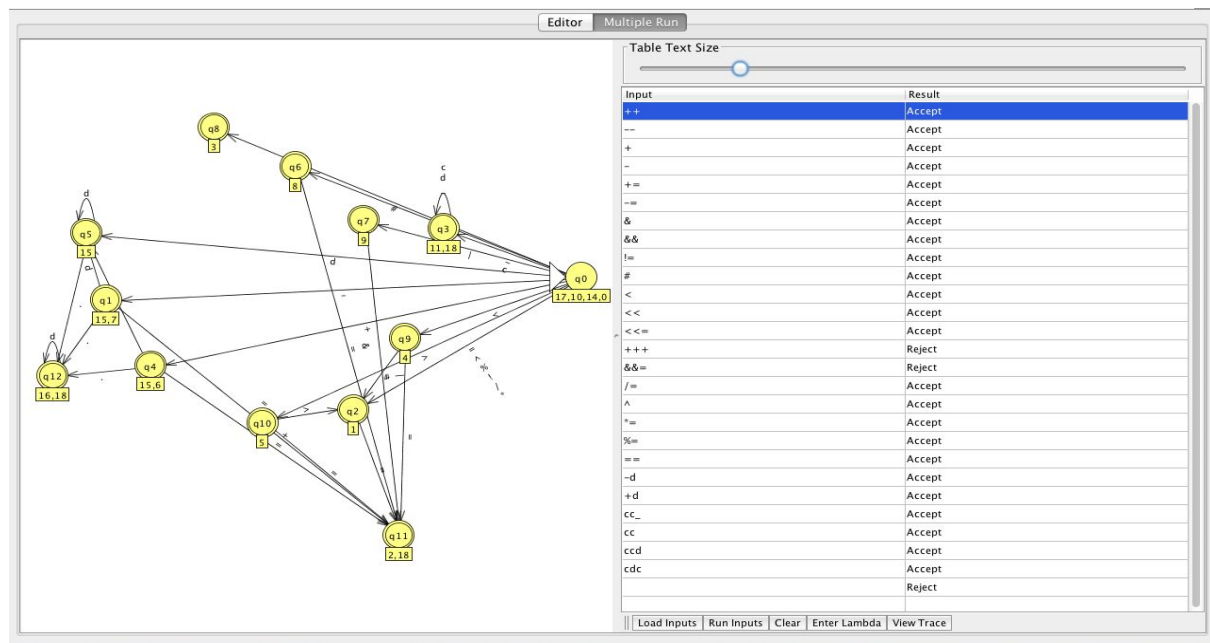
## 1.6 DFA



Legenda:

- a = todos os caracteres de A-Z a-z.
- d = todos os dígitos de 0-9.

## 1.7 Testes de simulação de aceitação e rejeição de cadeias

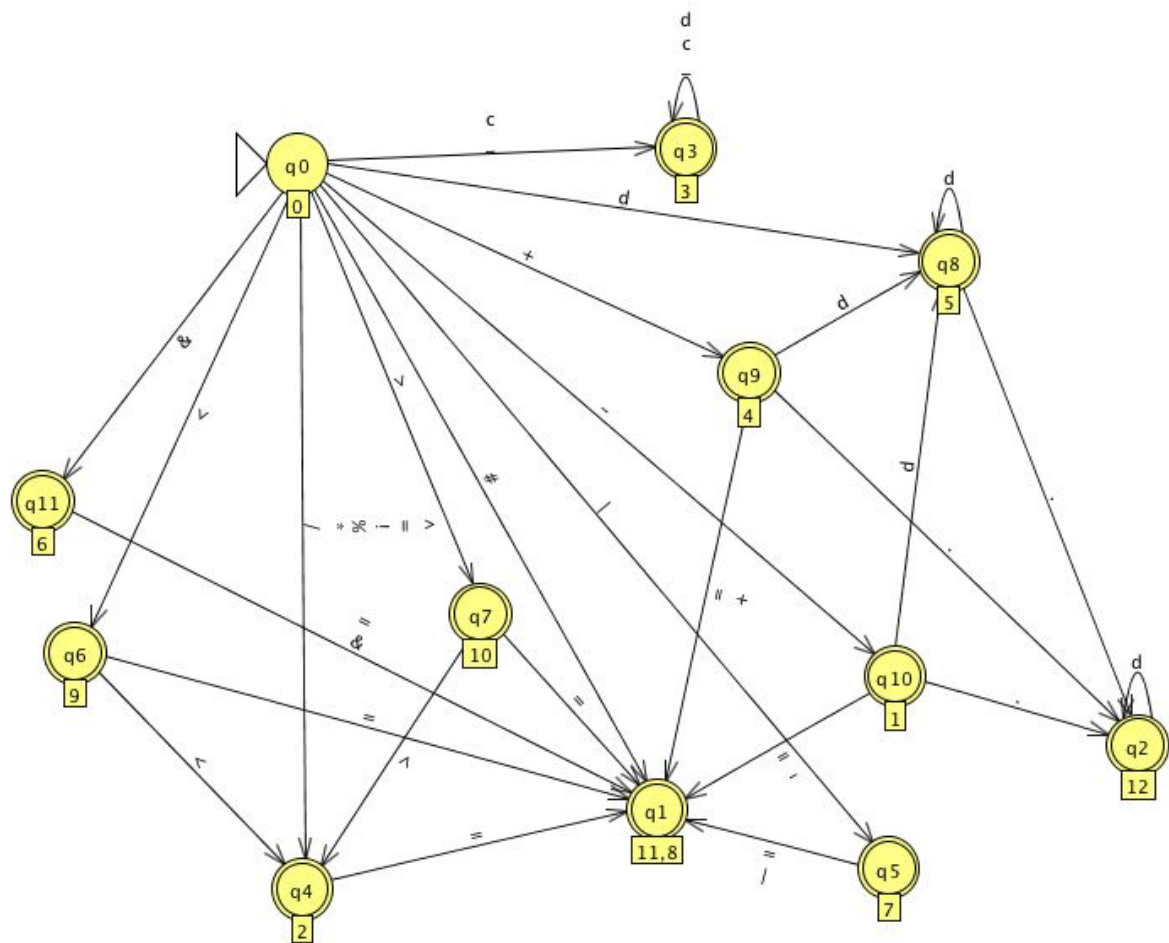


Legenda:

c = caracteres A-Z a-z

d = dígito 0-9

## 1.8 DFA Minimizado





## 1.9 Gramatica:

- $G = (\{A, S, I, E, H, D, B, L, C, G, K, F, J\}, \{A-Z, a-z, 0-9, \text{especiais}, \_ \}, P, S)$

especiais = caracteres especiais permitidos na linguagem C, Exemplo: ( =, ==, <, >, <=, >=, !=, &&, ||, \*=, +=, -=, /=, %=, etc).

- Regras de produção P:

$$S \rightarrow +E \mid >B \mid <H \mid /A \mid *A \mid \%A \mid !A \mid || \mid =A \mid \&J \mid dD \mid ^A \mid -C \mid cG \mid \_G \mid \#F$$
$$A \rightarrow =K \mid \varepsilon$$
$$I \rightarrow IK \mid \varepsilon \mid =K$$
$$E \rightarrow .L \mid \varepsilon \mid +K \mid dD$$
$$H \rightarrow <A \mid =K \mid \varepsilon$$
$$D \rightarrow .L \mid \varepsilon \mid dD$$
$$B \rightarrow >A \mid \varepsilon \mid =K$$
$$L \rightarrow \varepsilon \mid dL$$
$$C \rightarrow dD \mid .L \mid -K \mid \varepsilon \mid =K$$
$$G \rightarrow \varepsilon \mid \_G \mid cG \mid dG$$
$$K \rightarrow \varepsilon$$
$$F \rightarrow \varepsilon$$
$$J \rightarrow \varepsilon \mid =K \mid \&K$$

Legenda:

a = caracteres A-Z a-z

d = dígito 0-9

## 2 Expressões regulares (GREGP)

- a) '[A-Z-]+[ ]\*[A-Z]\*[ ][0-9]+[ ][Xx][ ][0-9]+[ ][A-Z-]+[ ]\*[A-Z]\*'

```
dorici@Argos:~/Formais$ grep -E '[A-Z-]+[ ]*[A-Z]*[ ][0-9]+[ ][Xx][ ][0-9]+[ ][A-Z-]+[ ]*[A-Z]*' noticias_grep.txt
CORINTHIANS 1 X 0 VASCO
GRÊMIO 0 X 1 CHAPECOENSE
VITÓRIA 1 X 2 SÃO PAULO
ATLÉTICO-PR 3 X 1 FLUMINENSE
PONTE PRETA 1 x 3 ATLÉTICO-GO
BOTAFOGO 2 X 0 SANTOS
PALMEIRAS 1 X 0 CORITIBA
dorici@Argos:~/Formais$
```

- b) '[0-9]{2}[ ]de[ ][A-Za-z]+[ ]de[ ][0-9]{4},?[ ]\([?[a-z-]+\)?'

```
dorici@Argos:~/Formais$ grep -E '[0-9]{2}[ ]de[ ][A-Za-z]+[ ]de[ ][0-9]{4},?[ ]\([?[a-z-]+\)?' noticias_grep.txt
Data: 17 de setembro de 2017, domingo
Data: 17 de setembro de 2017, domingo
Data: 17 de setembro de 2017, domingo
Data: 17 de setembro de 2017, domingo
Data: 16 de setembro de 2017, sábado
Data: 16 de setembro de 2017 (sábado)
Data: 18 de setembro de 2017, segunda-feira
dorici@Argos:~/Formais$
```

- c) 'Renda:[ ]R?\\$?[0-9]\*\.[0-9]+\.[0-9]+,[0-9]+'

```
dorici@Argos:~/Formais$ grep -E 'Renda:[ ]R?\$?[0-9]*\.[0-9]+\.[0-9]+,[0-9]+' noticias_grep.txt
Renda: 2.436.134,70
Renda: R$ 406.709,00
Renda: R$ 159.210,00
Renda: R$ 767.222,50
dorici@Argos:~/Formais$
```

- d) Não é possível encontrar apenas os nomes dos times separadamente pois não existe nada para diferenciá-los de outros nomes próprios e palavras que possuem apenas letras maiúsculas.

### Exemplos:

casos em que o nome começa com letra maiúscula seguido de n letras minúsculas:

**nome de time:** Botafogo

**nome de próprio:** (Carlos Eduardo)

**nome próprio seguido do time:** Marlon (Fluminense)

casos em que os nomes possuem todas as letras maiúsculas:

nomes de times: ATLÉTICO-PR, (PALMEIRAS), SÃO PAULO

demais palavras com todos os caracteres maiúsculos: FICHA TÉCNICA, CARTÕES AMARELOS,(AGO).

### 3 Gramáticas Regulares e Prolog

- **Gramática regular para gerar a seguinte receita de ovos cozidos:**

- Submerja os ovos
- Acrescente sal
- Tampe a panela
- Deixe ferver

```
P = {  
    Objetos -> Panela(com água), Ovos, Sal  
    Panela Ovos -> OvosNaPanela  
    OvosNaPanela Sal -> OvosNaPanelaComSal  
    OvosNaPanelaComSal -> OvosCozidos  
}
```

$G = ( \{ \text{Objetos, Panela Ovos, OvosNaPanela Sal, OvosNaPanelaComSal} \}, \{ \text{OvosCozidos} \}, P, \text{Objetos} )$

- **Gerando números inteiros em Prolog:**

- **Programa:**

```
proximo_inteiro(I) :-  
    proximo_inteiro(0, I).
```

```
proximo_inteiro(I, I).  
proximo_inteiro(I, J) :-  
    I2 is I + 1,  
    proximo_inteiro(I2, J).
```

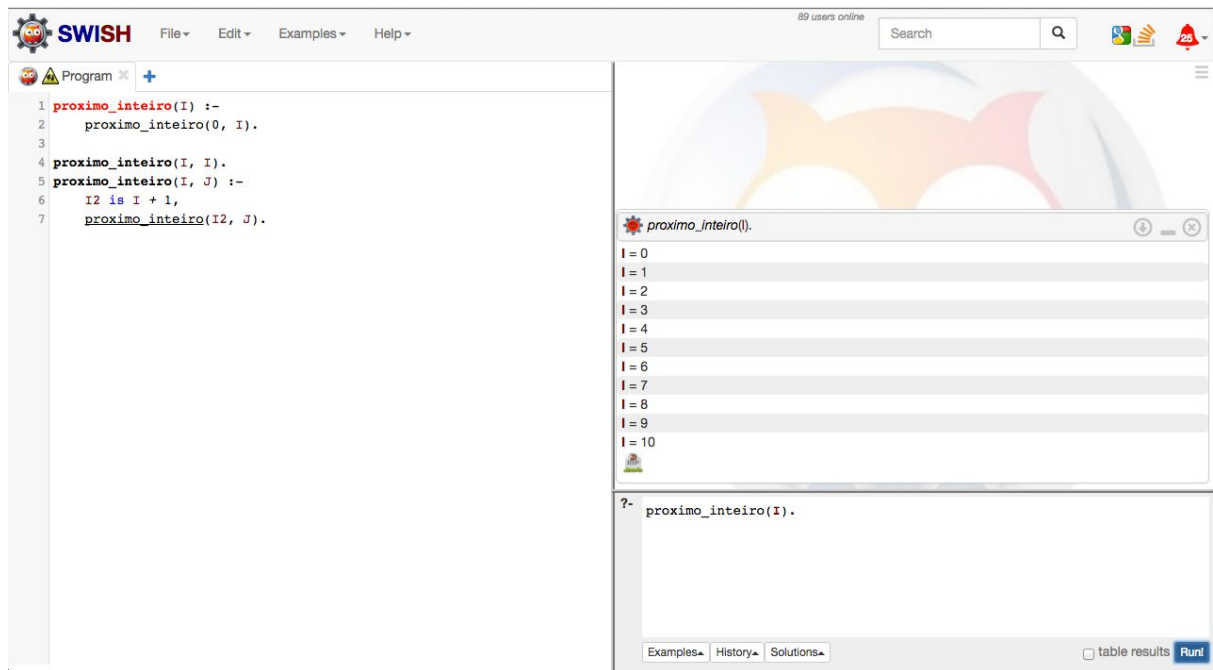
- **Consulta:**

```
proximo_inteiro(I).
```

## - Resultado:

# Através do programa acima, o prolog pode gerar quantos números inteiros for necessário, exemplo:

I = 0, I = 1, I = 2, I = 3, I = 4, I = 5, I = 6, I = 7, I = 8, I = 9, I = 10



## 4 Referências

- JFLAP Tutorial: [www.jflap.org/tutorial](http://www.jflap.org/tutorial)
- Linux Man Page “GREP”: <http://www.skrenta.com/rt/man/grep.1.html>
- Prolog Syntax: <http://www.swi-prolog.org/man/syntax.html>
- Conjunto de slides da disciplina “SCC-205 - Teoria da computação e linguagens formais”.